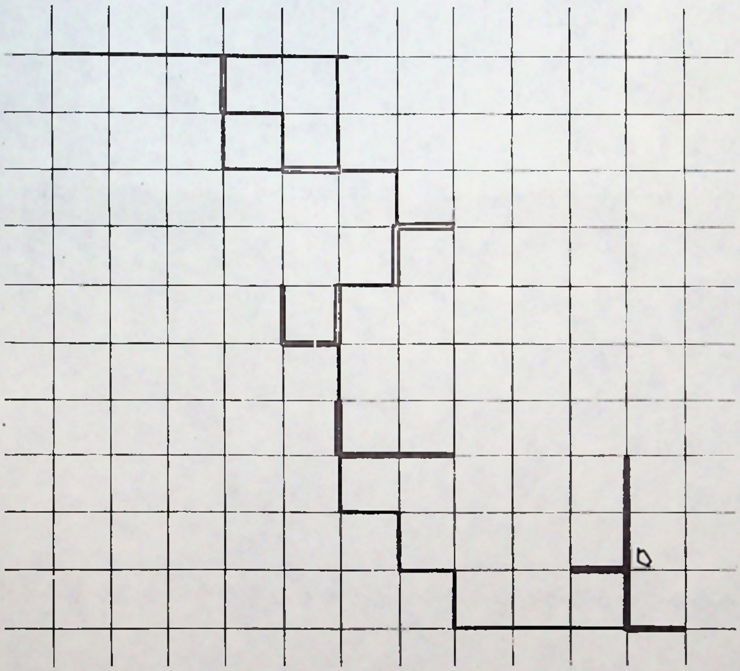
Popular Computing

The world's only magazine devoted to the art of computing.

Volume 9 Number 4

April 1981

97



A Numberprint For Pi

	1 4 1 5 9 2 6 5 3
0	2 8 3 1 8 5 3 0 6
0	5 6 6 3 7 0 6 1 2
1	1 3 2 7 4 1 2 2 4
0	2 6 5 4 8 2 4 4 8
0	5 3 0 9 6 4 8 9 6
1	0 6 1 9 2 9 7 9 2
0	1 2 3 8 5 9 5 8 4
0	2 4 7 7 1 9 1 6 8
0	4 9 5 4 3 8 3 3 6
0	9 9 0 8 7 6 6 7 2
1	9 8 1 7 5 3 3 4 4
1	9 6 3 5 0 6 6 8 8
1	9 2 7 0 1 3 3 7 6
1	8 5 4 0 2 6 7 5 2
1	7 0 8 0 5 3 5 0 4
1	4 1 6 1 0 7 0 0 8
0	8 3 2 2 1 4 0 1 6
1	6 6 4 4 2 8 0 3 2
1	3 2 8 8 5 6 0 6 4
0	6 5 7 7 1 2 1 2 8

S

An example of a scheme for converting a fraction (in this case, 9 digits of the mantissa of pi) from decimal to binary.

In each of the Numberprints shown, the origin is indicated with an arrow. The heavy lines indicate legs of the print that are traversed more than once.

Publisher: **Audrey Gruenberger**
Editor: **Fred Gruenberger**
Associate Editors: **David Babcock**, **Irwin Greenwald**, **Patrick Hall**
Contributing Editors: **Richard Andree**, **William C. McGee**, **Thomas R. Parkin**, **Edward Ryan**
Art Director: **John G. Scott**
Business Manager: **Ben Moore**

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $20.50 per year, or $17.50 if remittance accompanies the order. For Canada and Mexico, add $1.50 per year. For all other countries, add $3.50 per year. Back issues $2.50 each.

Numberprints

Suppose we take a number, like pi, and convert it from its familiar decimal form to binary. The rules for converting integers and fractions are different. In Figure S, a scheme for the conversion is shown. The fraction is doubled, and whenever the doubling spills over across the decimal point, a "1" is generated in the binary number, starting at the binary point; otherwise, a "0" is generated. In the illustration, the first 20 bits of pi are developed, which is about all that can be expected from a 9-digit approximation to pi.

We thus arrive at:

pi = 11.00100100001111110110101010001000...

Now, if we interpret these bits two at a time according to this scheme:

0 0 = North

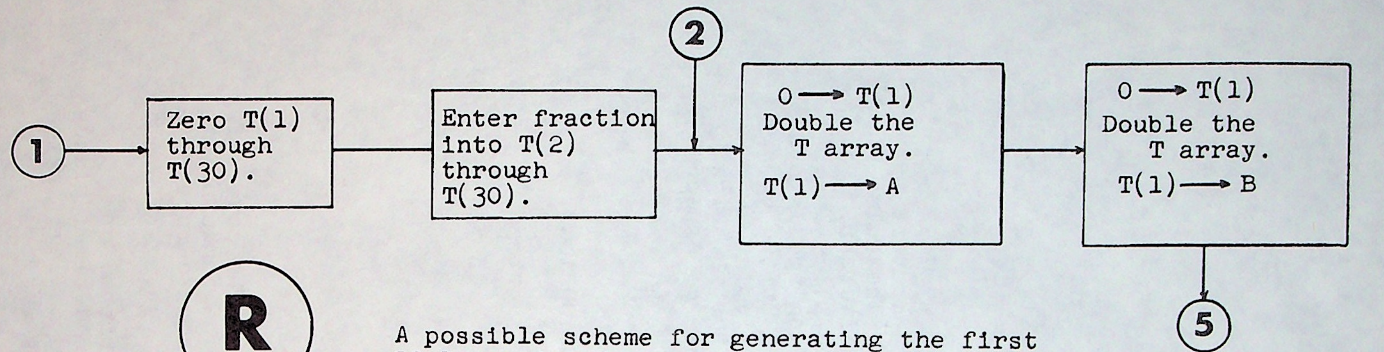
0 1 = East

1 0 = West

1 1 = South,

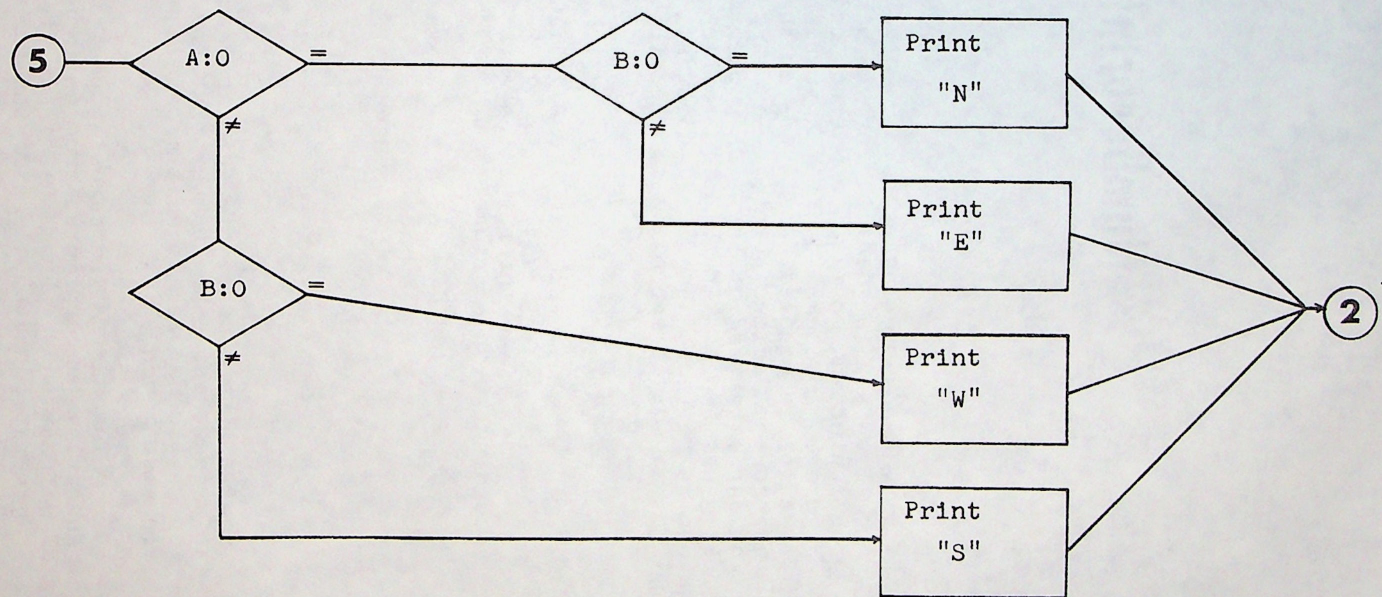
then we have a means of picturing numbers. The cover of this issue shows the Numberprint for pi. The Numberprints for e (the natural logarithm base) and the square root of 2 are also given.

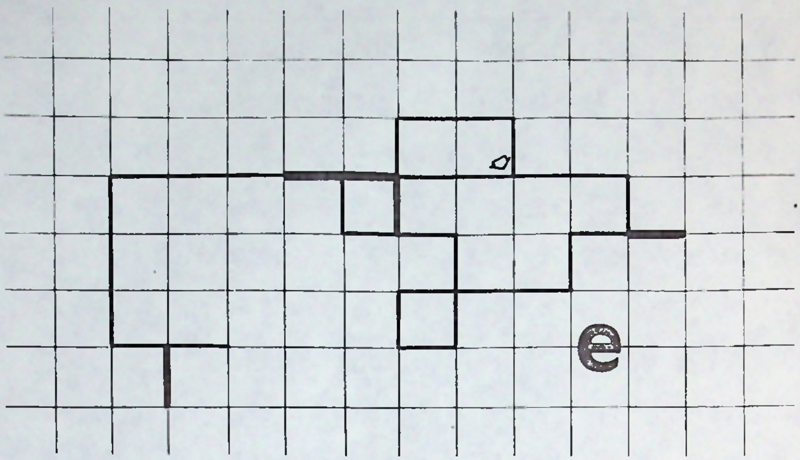
Flowchart R indicates a possible algorithm for generating Numberprints, from an input of 29 decimal places of a fraction. From this input, it should be legitimate to generate about 50 legs of the Numberprint.



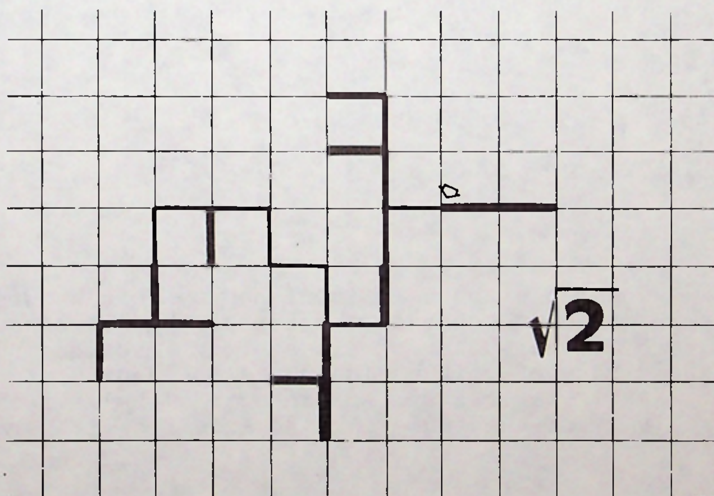
R

A possible scheme for generating the first 50 legs of a Numberprint.





Numberprints for e and the square root of 2.
The pattern is readily generated in any of
the BASICs, and could be produced dynamically
on a CRT display.



N	S	P
6	1.00000 00000 00000 00000	6.00000 00000 00000 00000
12	.51763 80902 05041 52470	6.21165 70824 60498 29637
24	.26105 23844 40103 18310	6.26525 72265 62476 39432
48	.13080 62584 60286 13363	6.27870 04060 93734 41427
96	.06543 81656 43552 28413	6.28206 39017 81019 27622
192	.03272 34632 52973 56329	6.28290 49445 70924 15090
384	.01636 22792 07874 25857	6.28311 52158 23715 29103
768	.00818 12080 80524 69579	6.28316 77842 96636 81734
1536	.00409 06125 82328 19023	6.28318 09264 56100 19148
3072	.00204 53073 60676 60908	6.28318 42119 98543 10109
6144	.00102 26538 14027 39500	6.28318 50333 84314 89519
12288	.00051 13269 23724 83463	6.28318 52387 30767 91038
24576	.00025 56634 63951 30948	6.28318 52900 67381 79334
49152	.00012 78317 32236 76626	6.28318 53029 01535 30341
98304	.00006 39158 66151 02207	6.28318 53061 10073 68338
196608	.00003 19579 33079 59090	6.28318 53069 12208 27853
393216	.00001 59789 66540 30543	6.28318 53071 12741 92733
786432	.00000 79894 83270 21647	6.28318 53071 62875 33953
1572864	.00000 39947 41635 11620	6.28318 53071 75408 69258
3145728	.00000 19973 70817 55910	6.28318 53071 78542 03084

The table above is furnished by John W. Wrench, Jr., to replace the one on the top of page 11 of our issue No. 12. N is the number of sides of an inscribed polygon in a circle of unit radius; S is the length of a side; P is the perimeter. Mr. Wrench says "I believe that the second part of the table is also infested with errors."

Ulam-3

In issues 13 and 19 there was discussion of a problem due to S. Ulam:

> In the sequence 1, 2, 3, 4, 6, 8, 11, 13,... each new member can be formed in one and only one way by adding two earlier numbers of the sequence.

The first 110 terms of this sequence were given, calculated by a program of Associate Editor David Babcock's.

The accompanying flowcharts show a scheme for extending the idea to numbers that can be formed in one and only one way by adding <u>three</u> earlier numbers.

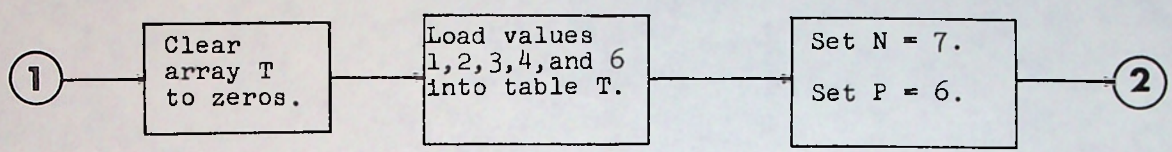
The first few terms of this sequence (call it Ulam-3) are given here:

1	2	3	4	6	7	8
19	21	22	23	38	58	59
74	77	91	92	108	124	125
127	163	178	198	213	230	<u>246</u>
<u>247</u>	<u>248</u>	<u>249</u>	265	266	282	283
299	316	336	337	352	369	371
386	404	406	422			

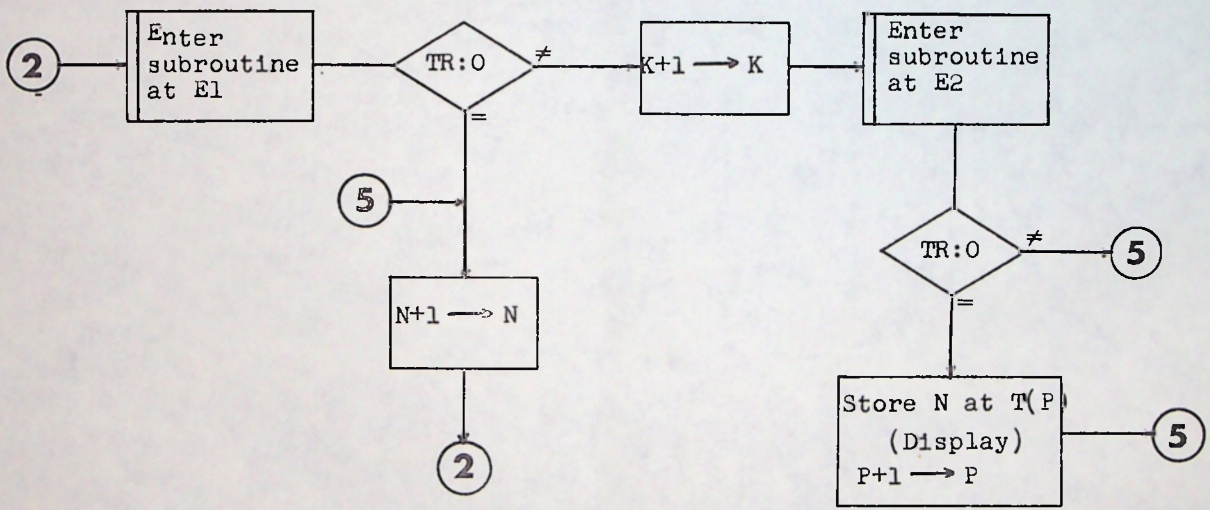
The 28th through 31st terms (underscored) are conjectured to be the only instance (other than the first four generating terms) of four consecutive numbers.

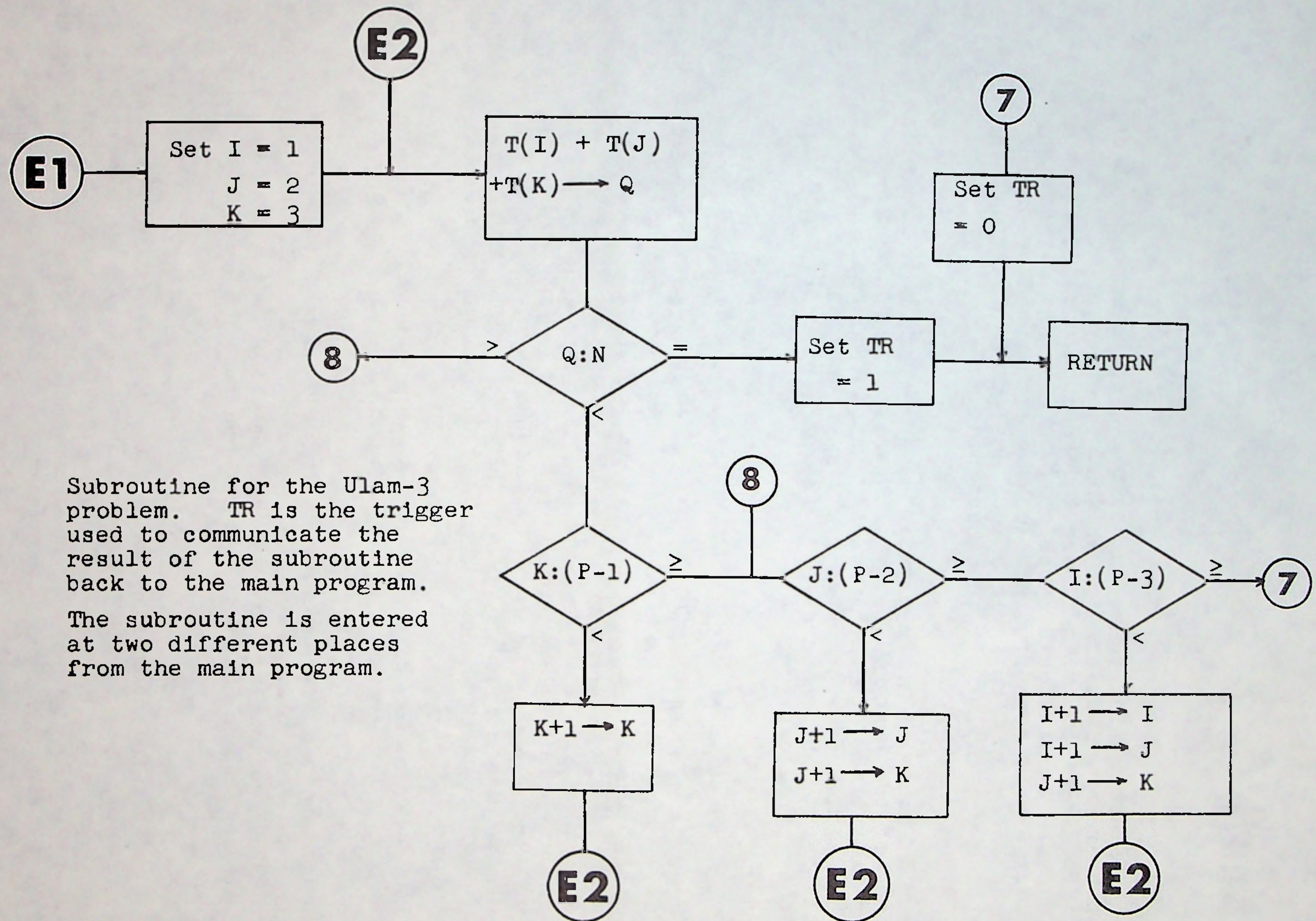
The problem lends itself to coding in almost any language (even integer BASIC will carry it quite far). The suggested solution, shown on the following two pages, slows down considerably as the numbers increase. Hence, to extend the sequence another 100 terms would require either a machine language approach or, more likely, one of those analytic solutions that brilliant problem solvers like David Ferguson keep coming up with.

PROBLEM 294



The logic of the Ulam-3 problem. N is used for consecutive integers being tested. P is the term number of the sequence being developed.





Subroutine for the Ulam-3 problem. TR is the trigger used to communicate the result of the subroutine back to the main program.

The subroutine is entered at two different places from the main program.

Knuth

One recurring theme in our pages has been the notion that there are problems for which the computer is the only tool for solution. There are indeed such problems (such as certain combinatorial problems) but many of our attempts to devise good computing problems were thwarted by having someone come up with an ingenious analytic solution, thus spoiling the fun. It just happened again:

In issue 95 there appeared a straightforward solution to one of the Penny Flipping problems, using an approach that lent itself to machine language on a microcomputer. This was clearly a computer problem and for most people it still is. Not, however, for the master; Prof. Don Knuth writes as follows:

When PC95 arrived last week I took another look at penny flipping ... and I got lucky!

Enclosed is an extension of "Table W". For each n I list the period length $p(n)$ when the penny flipping starts to repeat, and also any smaller number of steps where the coins all come out heads up but the last move was not to flip the whole stack. These latter "sporadic solutions" are rare, although they occur for infinitely many n; it isn't hard to prove that you get a sporadic solution of length $p(n) - 2k - 1$ whenever $n = 1 + 2 + \cdots + 2k = k(2k+1)$ and $k > 1$. All of the other sporadic solutions on this list are of the form $\frac{1}{2}\,p(n) - 1$; such solutions occur when the pennies in the second half of the period are complemented from their state in the first half of the period. It seems that there are two different sporadic solutions only when $n = 21$; but it will almost surely be impossible to prove this conjecture rigorously, or to prove that all sporadic solutions are of these two types.

I plan to give this nice problem to our grad students the next time I teach Stanford's programming seminar. It illustrates the way a good idea can speed up computations even for problems that appear to have no simple structure.

In fact, the use of some elementary ideas from arithmetic and group theory make it possible to compute Table W as fast as my hardcopy terminal could print it at 30cps, even when I was timesharing the computer with many other people. The case $n = 113$ required about .6 of a second, by comparison with the method of your program which would have taken more than two days.

The program that produced this table was about 150 lines of ALGOL code. As you know, I have nothing against assembly language programming—once I had to write some code that was to be executed a trillion times, so that saving 100 nanoseconds in the loop meant saving hours of expensive computer time—but sometimes it helps more to have a good idea. Yes, Euclid's algorithm pays off, even when flipping pennies.

It is clear, I think, that Prof. Knuth must have some very clever way of solving this problem, but we may never know what it is. He mentioned that his solution for N = 4096 took 25 minutes of CPU time. The method that appeared in issue 95 would require something on the order of 78 centuries.

Anyway, it is gratifying to find that not only has Prof. Knuth forgiven me for my blunder in issue 92, but is willing to contribute to the magazine once again.

But I can't resist the temptation to try, just once more, to develop a simple problem that must be solved by computer. Try this variation of Throwback (the original version was on the cover of issue 55):

Consider all the integers, starting with 3:

3 4 5 6 7 8 9 10 11 12 13 14 15...

The number at the head of the stream is called the leader. The leader is to be thrown back in the stream the number of positions given by its own value (that was the original problem) or twice that number of positions if it is prime. Thus, the first few moves are these:

4 5 6 7 8 9 3 10 11 12 13 14 15...

5 6 7 8 4 9 3 10 11 12 13 14 15...

6 7 8 4 9 3 10 11 12 13 5 14 15...

7 8 4 9 3 10 6 11 12 13 5 14 15...

and new leader values appear on this schedule:

First appearance of:	On move number:
3	0
4	1
5	2
6	3
7	4
8	5
9	7
10	9

When will the number 100 first appear as the leader?

Can this number be found without a computer?

(198607)

PROBLEM 295

1: 2
2: 8 3
3: 3
4: 24 11
5: 20
6: 36
7: 70
8: 112 55
9: 162 80
10: 300 295
11: 121
12: 240 119
13: 130
14: 1176
15: 300
16: 1440
17: 1428
18: 576
19: 2128
20: 320
21: 882 875 440
22: 968 483
23: 2622
24: 432
25: 1900
26: 780
27: 4320
28: 756
29: 4466
30: 26400
31: 11284
32: 2688
33: 20790
34: 28560
35: 16170
36: 16848 16839
37: 16872
38: 83980
39: 13104
40: 15960
41: 30668
42: 5880
43: 13330
44: 230384
45: 62100
46: 2484
47: 1410
48: 264960
49: 32634 16316
50: 12900
51: 27744
52: 2600
53: 2226
54: 1156680
55: 75900 75889
56: 11760
57: 13680
58: 493696
59: 14868
60: 140400
61: 14640
62: 135408
63: 232848
64: 51072 25535
65: 1034280
66: 54648
67: 67536
68: 138720
69: 8694 4346
70: 260680
71: 349888
72: 47520
73: 91980
74: 186480
75: 7182000
76: 20520
77: 5544
78: 28080 28067
79: 1161774
80: 336000
81: 5320728
82: 1883376
83: 68475
84: 1585836
85: 285090
86: 1475760
87: 456228
88: 2265120
89: 1226420
90: 75600
91: 301938
92: 380880
93: 539028
94: 78960
95: 1938000
96: 2428416
97: 30264
98: 19208 9603
99: 67320
100: 273000
101: 8243620
102: 1105272
103: 519120
104: 18258240
105: 2425500 2425485
106: 2281968
107: 38520
108: 21168
109: 5992602
110: 23760
111: 1210344
112: 2173248
113: 88750200
114: 105336
115: 425040
116: 341040
117: 2069496
118: 12198368
119: 68544
120: 13543200
121: 49832640
122: 717360
123: 127920
124: 11561760
125: 7980000
126: 89208
127: 748030
128: 2376192
129: 143190
130: 146900
131: 149340
132: 91872
133: 3527160
134: 7211880
135: 256397400
136: 228480 228463
137: 13941120
138: 139104
139: 36140
140: 399840
141: 473760
142: 161880
143: 2793648
144: 290304
145: 35090
146: 7901520
147: 141190560
148: 2051280
149: 2199240
150: 198000
151: 3234420
152: 139840
153: 136256904
154: 17109400
155: 39776100
156: 35865648
157: 414480
158: 853200
159: 11514780
160: 207486720
161: 41972700
162: 100440
163: 665705040
164: 3247200
165: 4643100
166: 1593600
167: 13916778
168: 3153024
169: 2413320
170: 437920
171: 805410 805391
172: 58480 29239
173: 4650240
174: 171216
175: 4395300
176: 658240
177: 5559216
178: 42207360
179: 2536072
180: 4773600
181: 230796720
182: 54853344
183: 192150
184: 2104118016
185: 1361600
186: 5468400
187: 4319700
188: 2320871280
189: 14277025980
190: 282156840
191: 7556724
192: 322560
193: 1621200
194: 939736
195: 2550600
196: 939624
197: 74143314
198: 25945920
199: 802368
200: 891480000
201: 8842392
202: 52940160
203: 97310080
204: 156143232
205: 6371400
206: 4756402392
207: 5738040
208: 29213184
209: 60969480
210: 438918480 438918459
211: 33359100
212: 21452704
213: 608725458 304362728
214: 179439000
215: 9876240
216: 6188832
217: 3098760
218: 364496
219: 7015008
220: 853723200
221: 704888340
222: 6717720
223: 187320
224: 963217920
225: 376200
226: 59018544
227: 26164020
228: 299592 149795
229: 6718860
230: 21307200
231: 255633840
232: 5554080
233: 147475020
234: 1432080
235: 3584220
236: 4663360
237: 8859060
238: 9611868
239: 44568720
240: 35006400
241: 5632170
242: 401329242600
243: 763408800
244: 404796
245: 2603370000
246: 6125400 3062699
247: 3230760
248: 184675680
249: 6294720
250: 3077000
251: 13599180
252: 589680
253: 999827664 999827641
254: 19431000
255: 44553600
256: 738017280
512: 571936768
1024: 43231518720
2048: 964767498240
4096: 1575568834560

Knuth's results for the Penny Flipping II problem

Fred Gruenberger

Swan Song

This is the last issue of POPULAR COMPUTING in its present form. The title will now be used by McGraw-Hill; the first issue with the new format by that group will emerge in October. We certainly wish them well.

I would like to summarize the adventures of publishing a highly specialized magazine for eight years.

What we had in mind when we started was a small business (actually, a self-supporting hobby) that would be run right; no customer would ever have legitimate grounds for complaint. Since the product is a magazine, the business is quite different from other businesses; for example, a magazine is the one product that people customarily pay for long in advance. We guess that, since we were dealing mainly with computer people, we could safely eliminate nonsense like "Renew now and get this FREE booklet." There is, of course, no way to tell whether or not this approach was the right one, but we felt that we had a fairly exact count of just how many people in the world were interested in our view of computing.

We ran no paid ads, and did little or no advertising ourselves. We figured that the word would spread by osmosis and indeed it did. A mention in Martin Gardner's column in Scientific American was a big help.

The main idea was to produce the kind of reading matter each month that I would like to read myself. Thus, "Letters to the editor" saying how wonderful the magazine is seemed inappropriate, space-wasting, self-serving, and--perhaps--redundant. "Articles" that were thinly disguised sales pitches were rejected, as were earth-shaking discoveries that computers can perform address modification.

We had our policies and our goals--the latter were spelled out in issue number 30--and they are still valid. Some of them were:

To encourage computing for fun.
To demonstrate <u>what</u> to compute as well as <u>how</u> to compute.
To emphasize the testing of programs.
To show that there is always a better way.
To call attention to the pitfalls and booby traps of our machines and languages.
To promote the art of computing.
To offer new problems for computer solution.

That last category will provide amusement for many years, as our original problems get picked up and reappear in other magazines and books.

We had to pick a level to aim at, somewhat above those to whom indexing and looping is a new and complicated notion, but below the level of the professionals who are creating our new systems and languages; it works out to about one semester of calculus. The mail was about equally divided between those who claimed they couldn't understand a word of it and those who objected to paying for such obvious trivia. In any event, we decided to leave it to others to teach elementary programming, rudimentary problem solving, and all dated information. With some exceptions, all the material that has appeared in the magazine has been timeless.

When producing a magazine more or less single-handed (in the first 96 issues, just 26% of the material was contributed by others), there are never any nagging worries about "what shall we do today?" As each issue is hauled to the postoffice, the pressure is already on to get cracking with the next issue. With this final issue, there have now appeared 1784 pages, of which 1320 pages were made up of original material written by me. Nearly 300 new computing problems were presented, with solutions for most of them. The names of some of these problems commemorate our travels (e.g., the Repulse Bay Trip, in issue 49).

The bragging above should not detract from the amazing contributions of others. David Babcock's brilliance influenced nearly every issue. We were proud to present the advanced thinking and competent writing of Richard Hamming, Thomas Parkin, and Daniel McCracken. Bill McGee's book reviews were a joy; he never tried to avoid the books that were difficult to review.

And, of course, the whole venture would have been impossible without the support, encouragement, and darn hard work of the cute publisher, whose idea it was in the first place.

When the magazine was started in 1973, many people took pains to convince us that we would run out of material within six months. In the eight years, there was exactly one month when we had to scramble for material. Most months there was a steady backlog of 40 to 60 pages of good stuff ready to go.

We had two slogans:

> "The way to learn computing is to compute"
>
> "The world's only magazine devoted to the art of computing"

For the latter slogan, we managed to devise some 70 distinct permutations (e.g., "The magazine for people who enjoy computing") and proceeded to run through them systematically on the covers. No one seemed to notice.

The magazine developed its ultimate character after about 12 issues. There were many appeals to change its character into something else; the most intense of these efforts was the bombardment by the APL fanatics, although in the last two years that has vanished (now it's the Pascal fanatics who are busy). I venture to predict that by 1982 another magic language will come along and we will hear all the old claims repeated again. Am I the only one who recalls going through the same charade with PL/I, FORTH, etc., etc.?

It took quite a while to get the printing problems licked. Issue number 6, for instance, was delayed when we found that the printer who held the masters had calmly closed his doors and gone fishing for a week. Issue number 66 has given everyone fits, since it was trimmed 1/4 inch larger in both dimensions than all other issues.

We tried our best to keep the literacy level up (you may have noticed that many articles in the current personal computing periodicals seem to be written by junior high school dropouts and show no evidence of any editing) and insure a decent standard of clear English. The keen eyes of our Associate Editors saved us from many a glitch.

After a magazine gets listed in standard reference works, its name gets on other lists, such as mailing lists, and these breed like rabbits. Not only do press releases arrive by the bale, but also regular issues of the Paint and Hardware Journal and ads for wire wrap machines, whatever they are. And among this detritus were the weekly letters that said "My teacher told us to write a paper on computers so please send all the information and hurry."--only those letters weren't spelled that well.

A magazine subscription list is an amazing thing to manage. Think about it: it is, by definition, highly volatile and full of unusual twists. Subscriptions that arrive from many foreign countries are widely separated in both time and space from the check that pays for them. Most company subscriptions arrive through an agency, and these subscriptions account for about 95% of all circulation problems. The communication channels in most large corporations are so tangled that the person who wants the magazine and requests it, and the agency clerk who relays the order, are poles apart. On one infamous subscription, we accumulated over an inch of correspondence getting it squared away.

We will not miss the monthly interface with the Postal Service (sic). Anyone who deals regularly with its inmates can recount tales of horror. When we began, in 1973, we paid 4.8¢ per copy and they did all the work; when we ended eight years later it was 8.4¢ per copy and we did all the work, under a set of rules you wouldn't believe. Each month the cost goes up in one way or another, the so-called service attenuates even more, and the rules of their game become ever more weird. One example will suffice. In our early years, we printed our envelopes with standard messages about "Return Postage Guaranteed" and "Change of Address Form Requested." We gave that up. In the last few years, when the address on a copy did not precisely match the occupant there, the postoffice charged us 28¢ to return the copy, and then only with the fascinating message "undelivered." What one wants, and what used to be supplied for 10¢, was the new address.

In our first two years, we received two flattering reviews in Computing Reviews, a publication of the Association for Computing Machinery. By dint of much howling and screaming, we managed to coerce them into a third review in 1978. Considering that POPULAR COMPUTING ran articles by two national presidents of ACM (neither of which they reviewed), not to mention the two marvelous articles by Don Knuth, one would think that a subsidized journal that purports to review the field would be able to manage more than three reviews in eight years.

The time frame for our magazine overlaps the deluge of personal computers and the emergence of some 30 or so periodicals that attempt to serve this new market. Along the way, several dozen magazines were born, got to issue 2 or 3, and quietly died. Even among those that have survived, none are edited by seasoned computer people. The various editors display astonishment (and, frequently, horror) at discovering that the number .1 just does not exist in binary; that programs can be and are copied; that computers can and do break down; that there is no upper bound to the number of BASIC renumbering programs that the world will buy; that unscrupulous people will try to pass off other people's work as their own; that

$$(\text{sine } x)^2 + (\text{cosine } x)^2$$

adds to unity for lots of values of x; that interactive programs to guess a number between 6 and 8 can be sold as "teaching tools" over and over. It seems that snake oil outsells medicine 100 to 1, as it always has.

To paraphrase Norman Sanders, it would be churlish of me not to acknowledge the contributions of California State University, Northridge, which provided me with many long committee meetings at which I could appear to be alert and attentive the while devising new computer problems.

A footnote to history: we determined, after exhaustive research, that Eagle "Prismacolor" Light Blue #904 pencils will disappear from Itek cameras and most photocopying machines better than any other such pencil.

Since there are always many folders lying around labelled "Work in progress," there is still a lot of unfinished business, as for example:

1. When and if Contributing Editor Thomas R. Parkin generates Part III of his definitive essay on Time (the first two parts were in issues 63 and 86), it may go unseen and enheralded, except by Tom and me. The main item left dangling is a magic formula that will convert dates from various irregular calendars (such as our present calendar) to the orderly scheme of one Joseph Scaliger. Many people, including competent astronomers, have tried to devise such a formula and failed, but Parkin cheerfully maintains that he can do it.

2. There was going to be an article about the unique features of the IBM 1620, and particularly its ingenious method of doing arithmetic by table lookup.

3. The work on placing isosceles triangles on a lattice (in issue 96) was intended to lead to a nice difficult computing problem.

4. Then there is the Foreign Exchange Problem. The daily paper lists the value of some 30 foreign currencies in terms of U.S. dollars (as on the cover of our issue number 83). For example, the rates for Peru and Israel have been:

	Peru	Israel
May 8, 1980	270.24	43.69
Aug 13, 1980	291.30	51.71
Sep 10, 1980	293.94	55.57
Nov 7, 1980	317.42	63.90
Jan 8, 1981	334.35	75.10
Mar 2, 1981	356.00	86.10

The number of units of the other currency that one dollar will buy.

(Israel's unit has been standardized to the early 1980 unit.)

Clearly, both currencies have been inflating, at least as compared to the dollar. But the dollar itself has been inflating, at a rate over 10% in the same time period. There should be a way to calibrate a currency on a more solid basis than its value in dollars (perhaps against a weighted total of all the other currencies), and therein might lie a computing problem.

5. Back in 1962 there was the PSAG project (the initials are those of Parkin, John Selfridge, George Armerding, and me). (Note: there are 24 possible permutations of the four letters, and we voted unanimously not to use either GASP or GAPS.) With advice and counsel from D. H. Lehmer, we had modestly set out to obsolete most of the number theory tables in the world. Specifically, we were going to calculate eight different functions for each of the first 6,000,000 primes (for example, the least exponent, the least positive primitive root, and so on). A test run was made at UCLA, which indicated that our logic was correct and that Armerding's program was debugged and tested. Selfridge had devised a scheme whereby the complete factorization of (p - 1) for each prime p could be coded into 5 characters, which would be invaluable when it came time to print out the 48,000,000 numbers we intended to produce. Parkin spent some happy hours with a punched card sorter, working out what we called Lehmer's Magic Numbers; they were the necessary conditions for small integers to be primitive roots, and they would save enormous amounts of computing time. A writeup on the PSAG project might have been interesting. To date, there has been no public outcry at the demise of the project.

6. There has never been a thorough description of the Mendenhall-Warren-Hollerith method of progressive digiting; this was a 1929 scheme for developing sums of products using punched card gear of the time. The method was widely used from 1929 to around 1948, during which time it was continually improved. With the introduction of various calculating devices (particularly the IBM 604), digiting fell into disuse. But it returned with a roar in the 1950's, as one of the methods for converting Hollerith information on punched cards into binary, for card entry to computers. In the 1960's, on computers using card input, more digiting was performed each hour than had been done in the whole world from 1929 to 1948.

This is typical of many things in computing; they go away for some time, only to reappear stronger than ever. (It may be safe to say that we will never go back to cathode ray tube storage.) The digiting process will probably be back again. In any event, it is a fascinating footnote to computing history, and it should be reported.

7. There are books on compiler writing, and even college courses in the study of compilers, but where is there any solid literature on interpreters? If we have some 500,000 personal machines today, increasing currently at around 200,000 per year, each of which executes BASIC or Pascal interpretively, then it would seem that interpreters would rate considerable attention. If everyone meekly settles for the sloppy programming represented by the BASIC interpreters now running in the popular personal machines (the world was using properly written interpretive systems before the kids who wrote those interpreters were born), then we are going backwards. The people who know how such things should be done have been singularly quiet.

8. The introduction of computers into cryptography, plus the recent attention given to public key systems, has directed all efforts toward substitution cipher systems. Little or no attention is paid to transposition ciphers, and they can be made quite secure. Moreover, they, too, lend themselves to computerization, and this matter should be looked into.

9. Since we fear no man nor beast, there was also going to be a critical article on the "Computing" merit badge of the Boy Scouts.

10. And, of course, there is a healthy backlog of new problems, solutions to old problems, and interesting correspondence. Perhaps some worthwhile articles may emerge from all this for the new POPULAR COMPUTING.

By and large, it's been fun. Beginning about two years ago, it began to resemble work, which both the publisher and I find cuts into the fun. It is time once more to exit gracefully, while the audience is still applauding.